ISBN: 972-8924-20-8 © 2006 IADIS

MODELO DE BENCHMARK PARA REPOSITÓRIOS DE DADOS DE SUPORTE A SERVIÇOS DE INFORMAÇÃO

José Novais
Universidade do Minho – Departamento de Sistemas de Informação
josenovais@josenovais.com

Leonel Duarte dos Santos
Universidade do Minho – Departamento de Sistemas de Informação
leonel@dsi.uminho.pt

RESUMO

A necessidade de fazer chegar informação a uma audiência cada vez mais vasta estimula o aparecimento de serviços com capacidades de recolha, armazenamento e difusão de informação, com o objectivo de facilitar a sua gestão, numa perspectiva de divulgação e de recolha, isto é, serviços de informação *online*. Para o bom funcionamento desses serviços, o armazenamento eficiente de informação assume um papel de grande relevância, pelo que se torna necessária a execução de testes para a selecção dos modelos mais adequados para esta tarefa. Neste trabalho são propostos um modelo e um sistema de testes que deverão ser capazes de permitir tirar conclusões sobre o modelo de armazenamento de informação a adoptar pelo repositório de dados de suporte a serviços de informação *online*.

PALAVRAS-CHAVE

Repositórios, *benchmarks*, bases de dados, serviços de informação *online*.

1. INTRODUÇÃO

Actualmente, com o aumento da utilização da Internet e das tecnologias como um meio de comunicação privilegiado e com a diversificação das aplicações que vão sendo colocadas *online*, tem-se assistido à disponibilização de cada vez maiores quantidades de informação a um cada vez maior número de utilizadores. Nesta perspectiva, as entidades vão disponibilizando serviços com capacidades de recolha, armazenamento e difusão de informação com vista a facilitar a gestão da informação, quer numa perspectiva de divulgação quer de recolha. Estes serviços implicam normalmente a manipulação de grandes quantidades de informação e de sistemas capazes de executar esta tarefa eficazmente, aos quais pode ser dado o nome de repositórios. Desta forma, os repositórios são o que alimenta os serviços de informação *online*.

Para o armazenamento de informação, e assumindo que tal é feito com recurso a bases de dados, existem actualmente dois modelos bastante divulgados: o relacional e XML. No que diz respeito ao armazenamento de XML, isto pode ser feito com recurso a bases de dados já existentes (por exemplo relacionais) às quais foram adicionadas características específicas para a sua manipulação – bases de dados com suporte para XML (*XML enabled*). Uma outra perspectiva para o armazenamento de XML em bases de dados é um novo tipo de base de dados, denominadas nativas XML. Neste tipo de bases de dados, a entrada, saída e armazenamento de informação é sempre no formato XML, não existindo qualquer tipo de conversões intermédias para outro formato. Apesar de ambas manipularem XML, as bases de dados XML nativas e *XML enabled* apresentam diferenças [Bourret 2003]:

- As nativas preservam a estrutura física dos documentos o que nem sempre é feito nas *XML enabled*.
- As nativas guardam qualquer ficheiro, mesmo sem conhecer o seu *schema*, o que pode não ser possível nas *XML enabled*.
- Nas nativas, a informação apenas é acessível em formato XML, ao passo nas *XML enabled* poderá existir outra forma de aceder à informação.

Com este artigo, pretende-se comparar os modelos para o armazenamento de informação referidos anteriormente, tendo sido feito para isso um conjunto de testes cujo modelo e resultado são apresentados.

O artigo está organizado da seguinte forma. Inicialmente são apresentadas ferramentas utilizadas para o teste de performance – *benchmarks*. De seguida, na secção 3 são definidos os cenários a considerar neste trabalho e um possível modelo para a execução de testes. Na secção 4 é descrito um sistema no qual serão conduzidos os testes. Por fim, na secção 5 apresentados resultados da execução dos testes e na secção 6 são delineadas as conclusões e o trabalho futuro.

2. BENCHMARKS

Existe uma grande diversidade de sistemas que manipulam e armazenam informação, baseada em diversos modelos como o relacional, orientado por objectos ou XML, com desempenhos e características variados. Desta forma, a necessidade de comparar desempenhos surge de uma forma natural.

Um *benchmark* é um programa utilizado para testar a performance de *software*, *hardware* ou um sistema [Collin 2002]. Mais concretamente, um *benchmark* para bases de dados pode ser visto como um conjunto de instruções utilizadas para medir e comparar o desempenho de dois ou mais sistemas de gestão de base de dados. Isto é feito recorrendo à execução de experiências bem definidas cujas medidas de desempenho serão usadas para prever o desempenho do sistema [Seng et al. 2005]. Desta forma, na especificação de um *benchmark* são considerados 3 componentes principais [Menascé 2002]: o sistema a ser testado (SUT – *System under test*), a carga de trabalho submetida ao SUT (*workload*), que consiste nas operações de teste, e uma ou mais métricas que são resultantes da monitorização e avaliação do desempenho do SUT o qual inclui a base de dados de teste. Exemplos de métricas são *throughput*, tempo de resposta, ou relação performance / custos de manutenção. Além disto, segundo [Gray 1993], um *benchmark* deve respeitar ainda princípios básicos bem definidos: relevância (deverá capturar as características do sistema a ser medido), portabilidade (deverá facilmente ser implementado em diferentes sistemas), escalabilidade (deverá ter a possibilidade de testar várias bases de dados em diferentes sistemas) e simplicidade (deverá ser perceptível).

Existem vários *benchmarks* utilizados em bases de dados relacionais, tais como Wisconsin [DeWitt 1993], AS^{3}AP [Turbyfill et al. 1989], mais orientados para o teste de aspectos importantes do servidor de base de dados, sendo constituídos por uma base de dados de teste e alguns *queries*. Outros *benchmarks* importantes são os definidos pelo *Transaction Processing Performance Council* (TPC), um consórcio de vários fabricantes de *software* e *hardware* sem fins lucrativos. Este consórcio define *benchmarks* para vários fins, como processamento de transacções *online* (OLTP), comércio electrónico, apoio à decisão, etc, e cujos resultados da utilização são publicados regularmente. Dois *benchmarks* definidos pelo TPC são o TPC-C [TPC 2005] e o TPC-W [TPC 2002], que têm grande aceitação na indústria.

A nível de bases de dados de XML, identificam-se dois tipos de *benchmarks*. Os *micro-benchmarks*, como o Michigan Benchmark [Runapongsa et al. 2003], são desenhados de forma a testarem componentes específicos do sistema com vista a isolar e ajudar a corrigir certos problemas. Pretendem explorar o impacto na performance do sistema das características mais importantes do XML, dispondo de uma base de dados de teste heterogénea, não inspirado numa qualquer aplicação real, sobre o qual são especificados *queries* especialmente desenhados para testar componentes elementares da linguagem de *query* (como selecção, *joins*, etc). Os *benchmarks* aplicacionais, por seu lado, funcionam a um nível mais elevado, pretendendo medir a performance do sistema como um todo e não questões específicas. Cada um deles dispõe de uma base de dados de teste, sobre a qual são definidos *queries* que pretendem abranger o maior número possível de características da linguagem de *query*. Exemplos destes benchmarks são o XOO7 [Li et al. 2001], Xmark [Schmidt et al. 2002], XBench [Yao et al. 2004] e o XMach-1 [Böhme e Rahm 2001].

3. CENÁRIOS E MODELO DE TESTES

A informação poderá estar modelada segundo o modelo relacional, devidamente inserida em registos de tabelas numa base de dados. No entanto, poder-se-á optar por a representar não utilizando o modelo relacional mas XML. A informação representada em XML poderá ser encarada de duas formas distintas: centrado nos dados ou centrado nos documentos. Apesar de não existirem regras rígidas, cada uma destas

hipóteses poderá estar mais adequada para utilização para diferentes tipos de bases de dados. Desta forma, uma visão centrada nos dados poderá ser mais adequada para utilização numa base de dados *XML enabled* ao passo que uma visão centrada no documento poderá ter melhor desempenho numa base de dados XML nativa [Vakali et al. 2005].

No contexto do presente trabalho, foram estudados cenários, numa perspectiva de divulgação de informação (isto é, consultas), onde foram utilizadas para o armazenamento físico de informação do repositório de suporte a um serviço de informação, bases de dados relacional (para a representação da informação no modelo relacional), XML *enabled* ou XML nativa (para suportar informação representada em XML). O estudo destes cenários deverá ter como base uma grande quantidade de informação de forma a atribuir o maior realismo possível ao estudo a efectuar. Desta forma, a quantidade de informação será outro aspecto com interesse no estudo, além do tipo de base de dados, podendo ser considerados cenários com quantidades distintas de informação.

Para tirar conclusões sobre os modelos mais adequados para a utilização em repositórios torna-se necessária a execução de testes, isto é de *benchmarking*. Torna-se necessário conceber e implementar um sistema, executar testes e interpretar os resultados. Trabalhos como [Böhme e Rahm 2001; Li et al. 2001; Schmidt et al. 2002; Yao et al. 2004] abordam questões relacionadas com o processamento de XML nomeadamente o processamento de *queries* que pretendem abranger o maior número possível de características da linguagem de *query*, pretendendo medir a performance do sistema como um todo, existindo um foco exclusivo em bases de dados de XML ao passo que outros como [Turbyfill et al. 1989; DeWitt 1993] abordam questões de processamento de *queries* em sistemas relacionais. Na presente situação o objectivo não é o mesmo que o dos trabalhos referidos uma vez que não se está particularmente interessado em explorar questões de processamento de XML ou características das linguagens de *query* ou ainda a capacidade de processamento destes pelos sistemas. A atenção foca-se antes na escolha de um modelo, XML ou relacional, que melhor responde às necessidades de um serviço de informação.

Podemos considerar que um serviço de informação com um perfil centrado na consulta de informação deverá permitir as seguintes funcionalidades básicas: acesso directo aos itens por chave (pesquisa por chave, obtenção de resultados ordenados segundo determinados critérios, obtenção de indicadores com base na informação armazenada, suporte eficiente para resultados de grandes dimensões (elevado número de itens) e finalmente pesquisas textuais. Com vista à realização de testes foram criadas pesquisas que implementam estas funcionalidades. Cada pesquisa tem 3 versões, uma para cada base de dados envolvida, na respectiva linguagem de *query* utilizada. Os modelos a testar são bastante diferentes, pelo que a implementação destas funcionalidades terá de abranger características existentes em ambos. A execução de testes pressupõe a medição de parâmetros relevantes. Um parâmetro importante é o *thoughput* que se pode definir como o número de pesquisas, baseadas nas características propostas, para o qual se obteve uma resposta por segundo.

4. SISTEMA DE TESTES

Para a execução de testes torna-se necessária a criação de um sistema que tenha em atenção as funcionalidades básicas descritas. A sua arquitectura é inspirada nas propostas em [Böhme e Rahm 2001; TPC 2002] sendo igualmente baseada numa aplicação *web*. É composta por três componentes: base de dados, servidor aplicacional e clientes, não existindo a necessidade do componentes adicionais externos aos SUT como o que faz recolha e *upload* de informação para a base de dados (*loader*) [Böhme e Rahm 2001] ou *gateway* de pagamentos [TPC 2002].

Na presente situação o sistema irá utilizar, em testes distintos, bases de dados para a manipulação de XML (*enabled* e nativa) e relacional, todas elas contendo informação equivalente representada nos respectivos modelos. A informação manipulada é referente a currículos de investigadores que inicialmente estava representada no modelo relacional, tendo sido convertida para XML. No servidor aplicacional é executada uma aplicação *web* com base num servidor *web* (*http*) bem como outros componentes de software necessários para acesso às bases de dados. O conjunto do servidor aplicacional e de base de dados constitui o sistema de teste, excluindo os clientes pelo que não se considera nos resultados dos testes atrasos devidos a comunicações e a processamentos feitos a nível dos clientes. Desta forma, os clientes conectam-se ao sistema de teste via o protocolo *http* e todos os resultados são medidos a nível do servidor aplicacional tendo apenas em conta o intervalo de tempo entre a chegada do pedido e o final do seu processamento (ignorando o tempo

que o resultado demoraria a chegar ao cliente). Para evitar atrasos relativos à rede, os servidores aplicacional e de bases de dados serão os mesmos, com dados armazenados localmente. Com vista a evitar inconsistências nos resultados, não é permitida *cache* ao nível do servidor. Para simular diferentes cenários são utilizadas diferentes bases de dados com diferente número de registos, bem com é alterado o número de clientes e consequentemente o número de pedidos ao servidor aplicacional. Ao *schema* da base de dados original foi feita uma simplificação, mas mantendo a estrutura básica de um currículo. O XML *Schema* foi criado com base no *schema* relacional simplificado. Apesar de terem sido utilizadas ferramentas automáticas para esta tarefa, a versão final teve uma revisão manual, tendo sido adoptada uma abordagem baseada em atributos para representar as colunas das tabelas e usando os nomes das colunas no nome dos atributos[Williams et al. 2000]. Após a criação do XML *Schema* foi feita a conversão da informação relacional para o formato XML, respeitando este *schema* e com recurso a ferramentas automáticas.

5. RESULTADOS DOS TESTES

Os testes realizados revelam (gráficos da Figura 1) que existem notoriamente 2 tipos de desempenho: alto para a base de dados relacional e baixo para as bases de dados de XML. A análise de variância dos resultados revela que os factores que mais influenciaram os resultados foram o tamanho da base de dados (33,52%) e o tipo (31,97%), ao passo que o número de clientes a influência foi menor (5,3%). Um outro aspecto que é observável nos valores obtidos é uma quebra acentuada de desempenho na base de dados de maiores dimensões, para todos os tipos. Mesmo assim a relacional está em vantagem. Esta quebra poderá ser devida a configurações do servidor onde foram executados os testes, pelo que estudos adicionais poderão de futuro clarificar este aspecto.

6. CONCLUSÃO E TRABALHO FUTURO

Neste documento foi introduzido um conceito de repositório bem como diferentes abordagens ao armazenamento de informação. Foi também proposto um modelo e um sistema de testes com os quais se pretende averiguar qual o modelo mais adequado para o armazenamento de informação nos repositórios. Para isto torna-se necessário a execução de testes, isto é, *benchmarking*. Estes testes terão de respeitar duas condições fundamentais para que possam ser credíveis. Deverão ter como base cenários de aplicação reais e não deverão abordar aspectos que não façam sentido em ambos os modelos considerados.

Os resultados obtidos nestes testes demonstram uma superioridade inegável da base de dados relacional. A escalabilidade de uma solução baseada no modelo relacional é bastante superior comparativamente com uma solução baseada em XML. Isto tem duas faces. Por um lado, as bases de dados relacionais apresentam décadas de desenvolvimento ao passo que a tecnologia XML é bastante mais recente. Desta forma, as bases de dados XML ainda não apresentam uma maturidade suficiente para possam ser colocadas ao nível das relacionais e para que o modelo XML possa ser uma alternativa ao modelo relacional. Por outro lado, a informação utilizada neste trabalho inicialmente estava numa base de dados relacional, tendo sido convertida para XML. Apresenta um perfil que se enquadra perfeitamente no modelo relacional, isto é, representação em tabelas e onde não existe uma noção de hierarquia. Estas situações são portanto mais favoráveis à utilização do modelo relacional [Lapis 2005], o que terá penalizado a base de dados de XML. É importante ter em atenção que existem situações onde uma base de dados de XML é mais apropriada, como gestão de informação orientada a documentos, integração de informação (integração entre aplicações, com informação obtida de diversas fontes), gestão de informação não estruturada ou evolução do *schema* [Bourret 2005]. Desta forma, seria bastante interessante repetir os testes deste trabalho mas com um tipo de informação mais apropriada ao modelo XML (com características como a possibilidade de alterações ao seu *schema* ou com hierarquias). Uma solução que adopte ambos os modelos aproveita o melhor de cada um deles: a rapidez associada ao modelo relacional (como demonstrado nos testes efectuados) e a capacidade de manipulação de documentos e lidar com *schemas* complexos ou que mudam frequentemente. Neste caso, parte da informação está armazenada em tabelas relacionais e outra parte em documentos XML, por exemplo a informação não estruturada cujo *schema* é pouco rígido e que poderá mudar mais frequentemente. Nesta perspectiva, o uso dos modelos relacional ou XML não pode ser visto numa perspectiva competitiva, mas sim complementar.

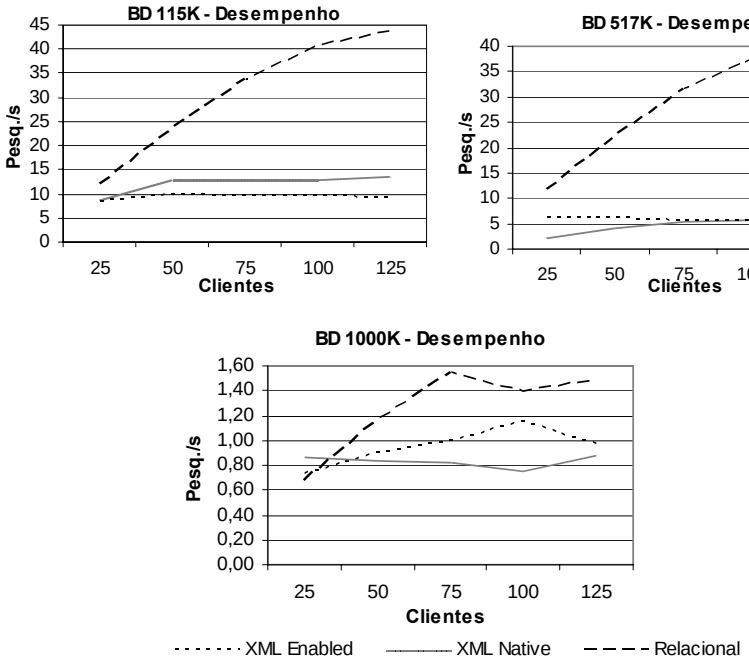

Figura 1: Resultados dos testes

REFERÊNCIAS

Böhme, T. and E. Rahm (2001). XMach-1: A Benchmark for XML Data Management. Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), Oldenburg, Germany, Springer-Verlag.

Bourret, R. (2003). "XML and Databases." Retrieved 15-1-2004, 2004, from http://www.rpbourret.com/xml/XMLAndDatabases.htm.

Bourret, R. (2005). Native XML Databases in the Real World. XML 2005 Conference & Exposition, Atlanta, USA.

Collin, S. M. H. (2002). Dictionary of Information Technology, Third Edition, Peter Collin Publishing.

DeWitt, D. J. (1993). The Wisconsin Benchmark: Past, Present, and Future. The Benchmark Handbook. J. Gray, Morgan Kaufmann Publishers, Inc.

Gray, J. (1993). The Benchmark Handbook, Morgan Kaufmann Publishers, Inc.

Lapis, G. (2005). XML and Relational Storage - Are they mutually exclusive? XTech 2005, Amsterdam, The Netherlands.

Li, Y. G., S. Bressan, et al. (2001). XOO7: Applying OO7 Benchmark to XML Query Processing Tools. Conference on Information and Knowledge Management, Atlanta, Georgia, USA, ACM Press, New York, NY, USA.

Menascé, D. A. (2002). TPC-W: A Benchmark for E-Commerce. IEEE Internet Computing. **6:** 83 - 87.

Runapongsa, K., J. M. Patel, et al. (2003). The Michigan Benchmark: Towards XML Query Performance Diagnostics. 29th VLDB Conference, Berlin, Germany.

Schmidt, A., F. Waas, et al. (2002). XMark: A benchmark for XML Data Management. 28th VLDB Conference, Hong Kong, China.

Seng, J.-L., S. B. Yao, et al. (2005). "Requirements-driven database systems benchmark method." Decision Suppot Systems 38(4): 629 - 648.

TPC (2002). Transaction Processing Performance Council - TPC Benchmark W, ver. 1.8. 2004.

TPC (2005). Transaction Processing Performance Council - TPC Benchmark C, ver. 5.4. 2005.

Turbyfill, C., C. Orji, et al. (1989). AS3AP:a comparative relational database benchmark. 34th IEEE Computer Society International Conference, San Francisco, CA.

Vakali, A., B. Catania, et al. (2005). "XML Data Stores: Emerging Practices." Internet Computing, IEEE 9(2): 62-69.

Williams, K., P. Dengler, et al. (2000). Professional XML Databases, Wrox Press Ltd.

Yao, B. B., M. T. Özsu, et al. (2004). XBench Benchmark and Performance Testing of XML DBMSs. 20th International Conference on Data Engineering, Boston, Massachusetts, U.S.A., IEEE Computer Society.